Probabilistic ODE Solvers with Runge-Kutta Means

Michael Schober  
MPI for Intelligent Systems  
Tübingen, Germany  
mschober@tue.mpg.de

David Duvenaud  
Department of Engineering  
Cambridge University  
dkd23@cam.ac.uk

Philipp Hennig  
MPI for Intelligent Systems  
Tübingen, Germany  
phennig@tue.mpg.de

Abstract

Runge-Kutta methods are the classic family of solvers for ordinary differential equations (ODEs), and the basis for the state of the art. Like most numerical methods, they return point estimates. We construct a family of probabilistic numerical methods that instead return a Gauss-Markov process defining a probability distribution over the ODE solution. In contrast to prior work, we construct this family such that posterior means match the outputs of the Runge-Kutta family exactly, thus inheriting their proven good properties. Remaining degrees of freedom not identified by the match to Runge-Kutta are chosen such that the posterior probability measure fits the observed structure of the ODE. Our results shed light on the structure of Runge-Kutta solvers from a new direction, provide a richer, probabilistic output, have low computational cost, and raise new research questions.

1 Introduction

Differential equations are a basic feature of dynamical systems. Hence, researchers in machine learning have repeatedly been interested in both the problem of inferring an ODE description from observed trajectories of a dynamical system [1, 2, 3, 4], and its dual, inferring a solution (a trajectory) for an ODE initial value problem (IVP) [5, 6, 7, 8]. Here we address the latter, classic numerical problem. Runge-Kutta (RK) methods [9, 10] are standard tools for this purpose. Over more than a century, these algorithms have matured into a very well-understood, efficient framework [11].

As recently pointed out by Hennig and Hauberg [6], since Runge-Kutta methods are linear extrapolation methods, their structure can be emulated by Gaussian process (GP) regression algorithms. Such an algorithm was envisioned by Skilling in 1991 [5], and the idea has recently attracted both theoretical [8] and practical [6, 7] interest. By returning a posterior probability measure over the solution of the ODE problem, instead of a point estimate, Gaussian process solvers extend the functionality of RK solvers in ways that are particularly interesting for machine learning. Solution candidates can be drawn from the posterior and marginalized [7]. This can allow probabilistic solvers to stop earlier, and to deal (approximately) with probabilistically uncertain inputs and problem definitions [6]. However, current GP ODE solvers do not share the good theoretical convergence properties of Runge-Kutta methods. Specifically, they do not have high polynomial order, explained below.

We construct GP ODE solvers whose posterior mean functions exactly match those of the RK families of first, second and third order. This yields a probabilistic numerical method which combines the strengths of Runge-Kutta methods with the additional functionality of GP ODE solvers. It also provides a new interpretation of the classic algorithms, raising new conceptual questions.

While our algorithm could be seen as a “Bayesian” version of the Runge-Kutta framework, a philosophically less loaded interpretation is that, where Runge-Kutta methods fit a single curve (a point estimate) to an IVP, our algorithm fits a probability distribution over such potential solutions, such that the mean of this distribution matches the Runge-Kutta estimate exactly. We find a family of models in the space of Gaussian process linear extrapolation methods with this property, and select a member of this family (fix the remaining degrees of freedom) through statistical estimation.
Table 1 lists all consistent methods of order $p \leq 3$ and number of stages $s = p$ (see [11]).

2 Background

An ODE Initial Value Problem (IVP) is to find a function $x(t) : \mathbb{R} \rightarrow \mathbb{R}^N$ such that the ordinary differential equation $\dot{x} = f(x, t)$ (where $\dot{x} = \partial x/\partial t$) holds for all $t \in T = [t_0, t_H]$, and $x(t_0) = x_0$. We assume that a unique solution exists. To keep notation simple, we will treat $x$ as scalar-valued; the multivariate extension is straightforward (it involves $N$ separate GP models, explained in supp.).

Runge-Kutta methods\(^1\) [9, 10] are carefully designed linear extrapolation methods operating on small contiguous subintervals $[t_n, t_{n+1}] \subset T$ of length $h$. Assume for the moment that $n = 0$. Within $[t_0, t_0 + h]$, an RK method of stage $s$ collects evaluations $y_i = f(\hat{x}_i, t_0 + hc_i)$ at $s$ recursively defined input locations, $i = 1, \ldots, s$, where $\hat{x}_i$ is constructed linearly from the previously-evaluated $y_{j<i}$ as

$$\hat{x}_i = x_0 + h \sum_{j=1}^{i-1} w_{ij} y_j, \quad (1)$$

then returns a single prediction for the solution of the IVP at $t_0 + h$, as $\hat{x}(t_0 + h) = x_0 + h \sum_{i=1}^{s} b_i y_i$ (modern variants can also construct non-probabilistic error estimates, e.g. by combining the same observations into two different RK predictions [12]). In compact form,

$$y_i = f \left( x_0 + h \sum_{j=1}^{i-1} w_{ij} y_j, \ t_0 + hc_i \right), \quad i = 1, \ldots, s, \quad \hat{x}(t_0 + h) = x_0 + h \sum_{i=1}^{s} b_i y_i. \quad (2)$$

$\hat{x}(t_0 + h)$ is then taken as the initial value for $t_1 = t_0 + h$ and the process is repeated until $t_n + h \geq t_H$.

A Runge-Kutta method is thus identified by a lower-triangular matrix $W = \{w_{ij}\}$, and vectors $c = [c_1, \ldots, c_s]$, $b = [b_1, \ldots, b_s]$, often presented compactly in a Butcher tableau [13]:

$$
\begin{array}{c|ccc}
   c_1 & 0 & & \\
   c_2 & w_{21} & 0 & \\
   c_3 & w_{31} & w_{32} & 0 \\
   \vdots & \vdots & \vdots & \ddots & \ddots \\
   c_s & w_{s1} & w_{s2} & \cdots & w_{s,s-1} & 0 \\
   \frac{1}{b_1} & \frac{1}{b_2} & \cdots & \frac{1}{b_{s-1}} & \frac{1}{b_s} \\
\end{array}
$$

As Hennig and Hauberg [6] recently pointed out, the linear structure of the extrapolation steps in Runge-Kutta methods means that their algorithmic structure, the Butcher tableau, can be constructed naturally from a Gaussian process regression method over $x(t)$, where the $y_i$ are treated as “observations” of $\hat{x}(t_0 + hc_i)$ and the $\hat{x}_i$ are subsequent posterior estimates (more below). However, proper RK methods have structure that is not generally reproduced by an arbitrary Gaussian process prior on $x$: Their distinguishing property is that the approximation $\hat{x}$ and the Taylor series of the true solution coincide at $t_0 + h$ up to the $p$-th term—their numerical error is bounded by $||x(t_0 + h) - \hat{x}(t_0 + h)|| \leq K h^{p+1}$ for some constant $K$ (higher orders are better, because $h$ is assumed to be small). The method is then said to be of order $p$ [11]. A method is consistent, if it is of order $p = s$. This is only possible for $p < 5$ [14, 15]. There are no methods of order $p > s$. High order is a strong desideratum for ODE solvers, not currently offered by Gaussian process extrapolators.

Table 1 lists all consistent methods of order $p \leq 3$ where $s = p$. For $s = 1$, only Euler’s method (linear extrapolation) is consistent. For $s = 2$, there exists a family of methods of order $p = 2$, parametrized

\(^1\)In this work, we only address so-called explicit RK methods (shortened to “Runge-Kutta methods” for simplicity). These are the base case of the extensive theory of RK methods. Many generalizations can be found in [11]. Extending the probabilistic framework discussed here to the wider Runge-Kutta class is not trivial.
by a single parameter $\alpha \in (0,1]$, where $\alpha = 1/2$ and $\alpha = 1$ mark the midpoint rule and Heun’s method, respectively. For $s = 3$, third order methods are parameterized by two variables $u, v \in (0,1]$.

**Gaussian processes (GPs)** are well-known in the NIPS community, so we omit an introduction. We will use the standard notation $\mu : \mathbb{R} \rightarrow \mathbb{R}$ for the mean function, and $k : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ for the covariance function; $k_{UV}$ for Gram matrices of kernel values $k(u_i, v_j)$, and analogous for the mean function: $\mu_T = \{\mu(t_1), \ldots, \mu(t_N)\}$. A GP prior $p(x) = \mathcal{GP}(x; \mu, k)$ and observations $(T, Y) = \{(t_1, y_1), \ldots, (t_s, y_s)\}$ having likelihood $\mathcal{N}(Y; x_T, \Lambda)$ give rise to a posterior $\mathcal{GP}^*(x; \mu^*, k^*)$ with

$$
\mu^*_s = \mu + k_{sT}(k_{TT} + \Lambda)^{-1}(Y - \mu_T) \quad \text{and} \quad k^*_{uv} = k_{uv} - k_{uT}(k_{TT} + \Lambda)^{-1}k_{Tv}.
$$

GPs are closed under linear maps. In particular, the joint distribution over $x$ and its derivative is

$$
p\left(\left[\begin{array}{c} x \\ \dot{x} \end{array}\right]\right) = \mathcal{GP}\left(\left[\begin{array}{c} x \\ \dot{x} \end{array}\right]; \left[\begin{array}{c} \mu \\ \dot{\mu} \end{array}\right], \left[\begin{array}{cc} k & k_y \\ k_y & k_y^\prime \end{array}\right]\right)
$$

with

$$
\mu^\alpha = \frac{\partial \mu(t)}{\partial t}, \quad k^\alpha = \frac{\partial k(t, t')}{\partial t'}, \quad \dot{\mu} = \frac{\partial \mu(t, t')}{\partial t}, \quad \dot{k}^\alpha = \frac{\partial^2 k(t, t')}{\partial t \partial t'}.
$$

A recursive algorithm analogous to RK methods can be constructed [5, 6] by setting the prior mean to the constant $x_0$, then recursively estimating $\hat{x}$ in some form from the current posterior over $x$. The choice in [6] is to set $\hat{x}_i = \mu^\alpha(t_0 + h c_i)$. “Observations” $y_i = f(\hat{x}_i, t_0 + h c_i)$ are then incorporated with likelihood $p(y_i | x) = \mathcal{N}(y_i; \hat{x}_i(t_0 + h c_i), \Lambda)$. This recursively gives estimates

$$
\hat{x}(t_0 + h c_i) = x_0 + \sum_{j=1}^{s-i} \sum_{\ell=1}^{i-j} k^\alpha(t_0 + h c_i, t_0 + h c_j) (\partial K^\alpha + \Lambda)^{-1} y_j = x_0 + h \sum_j w_{ij} y_j,
$$

with $\partial K^\alpha_{ij} = \partial^2 k(t_0 + h c_i, t_0 + h c_j)$. The final prediction is the posterior mean at this point:

$$
\hat{x}(t_0 + h) = x_0 + \sum_{i=1}^{s} \sum_{j=1}^{i-1} k^\alpha(t_0 + h, t_0 + h c_j) (\partial K^\alpha + \Lambda)^{-1} y_i = x_0 + h \sum_i b_i y_i.
$$

### 3 Results

The described GP ODE estimate shares the algorithmic structure of RK methods (i.e. they both use weighted sums of the constructed estimates to extrapolate). However, in RK methods, weights and evaluation positions are found by careful analysis of the Taylor series of $f$, such that low-order terms cancel. In GP ODE solvers they arise, perhaps more naturally but also with less structure, by the choice of the $c_i$ and the kernel. In previous work [6, 7], both were chosen ad hoc, with no guarantee of convergence order. In fact, as is shown in the supplements, the choices in these two works—square-exponential kernel with finite length-scale, evaluations at the predictive mean—do not even give the first order convergence of Euler’s method. Below we present three specific regression models based on integrated Wiener covariance functions and specific evaluation points. Each model is the improper limit of a Gauss-Markov process, such that the posterior distribution after $s$ evaluations is a proper Gaussian process, and the posterior mean function at $t_0 + h$ coincides exactly with the Runge-Kutta estimate. We will call these methods, which give a probabilistic interpretation to RK methods and extend them to return probability distributions, Gauss-Markov-Runge-Kutta (GMRK) methods, because they are based on Gauss-Markov priors and yield Runge-Kutta predictions.

#### 3.1 Design choices and desiderata for a probabilistic ODE solver

Although we are not the first to attempt constructing an ODE solver that returns a probability distribution, open questions still remain about what, exactly, the properties of such a probabilistic numerical method should be. Chkrebtii et al. [8] previously made the case that Gaussian measures are uniquely suited because solution spaces of ODEs are Banach spaces, and provided results on consistency. Above, we added the desideratum for the posterior mean to have high order, i.e. to reproduce the Runge-Kutta estimate. Below, three additional issues become apparent:

**Motivation of evaluation points** Both Skilling [5] and Hennig and Hauuberg [6] propose to put the “nodes” $\hat{x}(t_0 + h c_i)$ at the current posterior mean of the belief. We will find that this can be made
consistent with the order requirement for the RK methods of first and second order. However, our third-order methods will be forced to use a node \( \hat{x}(t_0 + hc) \) that, albeit lying along a function \( w(t) \) in the reproducing kernel Hilbert space associated with the posterior GP covariance function, is not the mean function itself. It will remain open whether the algorithm can be amended to remove this blemish. However, as the nodes do not enter the GP regression formulation, their choice does not directly affect the probabilistic interpretation.

**Extension beyond the first extrapolation interval** Importantly, the Runge-Kutta argument for convergence order only holds strictly for the first extrapolation interval \([t_0, t_0 + h]\). From the second interval onward, the RK step solves an estimated IVP, and begins to accumulate a global estimation error not bounded by the convergence order (an effect termed “Lady Windermere’s fan” by Wanner [16]). Should a probabilistic solver aim to faithfully reproduce this imperfect chain of RK solvers, or rather try to capture the accumulating global error? We investigate both options below.

**Calibration of uncertainty** A question easily posed but hard to answer is what it means for the probability distribution returned by a probabilistic method to be well calibrated. For our Gaussian case, requiring RK order in the posterior mean determines all but one degree of freedom of an answer. The remaining parameter is the output scale of the kernel, the “error bar” of the estimate. We offer a relatively simple statistical argument below that fits this parameter based on observed values of \( f \).

We can now proceed to the main results. In the following, we consider extrapolation algorithms based on Gaussian process priors with vanishing prior mean function, noise-free observation model (\( \Lambda = 0 \) in Eq. (3)). All covariance functions in question are integrals over the kernel \( k^0(i, i') = \sigma^2 \min(t - \tau, t' - \tau) \) (parameterized by scale \( \sigma^2 > 0 \) and off-set \( \tau \in \mathbb{R} \); valid on the domain \( t, t' > \tau \)), the covariance of the Wiener process [17]. Such integrated Wiener processes are Gauss-Markov processes, of increasing order, so inference in these methods can be performed by filtering, at linear cost [18]. We will use the shorthands \( t = t - \tau \) and \( t' = t' - \tau \) for inputs shifted by \( \tau \).

### 3.2 Gauss-Markov methods matching Euler’s method

**Theorem 1.** The once-integrated Wiener process prior \( p(x) = \mathcal{GP}(x; 0, k^1) \) with

\[
k^1(t, t') = \int_i^{i,t'} k^0(u, v)du dv = \sigma^2 \left( \frac{\min^3(t, t')}{3} + |t - t'| \frac{\min^2(t, t')}{2} \right)
\]

choosing evaluation nodes at the posterior mean gives rise to Euler’s method.
As theorem 3.

Consider the thrice-integrated Wiener process prior (The twice-integrated Wiener process is a proper Gauss-Markov process for all finite values of \( \tau \)). Choosing evaluation nodes at the posterior mean gives rise to the RK family of second order methods.

Theorem 2.

The theorem below shows that this only works after moving the onset \( h \) a departure from the policy of placing extrapolation nodes at the posterior mean.

Moving from second to third order, additionally to the limit towards an improper prior, also requires the final-step posterior covariance. Its finite values mean that this posterior indeed defines a proper GP.

3.3 Gauss-Markov methods matching all Runge-Kutta methods of second order

Extending to second order is not as straightforward as integrating the Wiener process a second time. The theorem below shows that this only works after moving the onset \( \tau \) of the process towards infinity. Fortunately, this limit still leads to a proper posterior probability distribution.

Theorem 2. Consider the twice-integrated Wiener process prior \( p(x) = GP(x; 0, k^2) \) with

\[
k^2(t, t') = \int_{\tau}^t \int_{\tau}^{t'} k^1(u, v) du dv = \sigma^2 \left( \frac{\min^5(t, t')}{20} + \frac{|t - t'|}{12} \left( \min^3(t, t') - \frac{\min^4(t, t')}{2} \right) \right).
\]

Choosing evaluation nodes at the posterior mean gives rise to the RK family of second order methods in the limit of \( \tau \to \infty \).

(11)

(12)

(13)
We note in passing that Eq. (15) vanishes for
This repeated integration simultaneously adds to and weakens certain prior assumptions in the
The above theorems have shown that the first three families of Runge-Kutta methods can be con-
This means that the final RK evaluation node does not lie at the posterior mean of the regressor.
Evaluating twice at the posterior mean and a third time at a specific element of the posterior
is intriguing that higher convergence order results from repeated integration of Wiener processes.
We close the main results by highlighting some non-obvious aspects. First, it
Some observations We close the main results by highlighting some non-obvious aspects. First, it
is intriguing that higher convergence order results from repeated integration of Wiener processes.
This repeated integration simultaneously adds to and weakens certain prior assumptions in the
implicit (improper) Wiener prior: \( s \)-times integrated Wiener processes have marginal variance
\( k^s(t, t) \propto t^{2s+1} \). Since many ODEs (e.g. linear ones) have solution paths of values \( \mathcal{O}(\exp(t)) \), it is tempting to wonder whether there exists a limit process of “infinitely-often integrated” Wiener processes giving natural coverage to this domain (the results on a linear ODE in Figure 1 show how the polynomial posteriors cannot cover the exponentially diverging true solution). In this context,
it is also noteworthy that $s$-times integrated Wiener priors incorporate the lower-order results for $s' < s$, so “highly-integrated” Wiener kernels can be used to match finite-order Runge-Kutta methods. Simultaneously, though, sample paths from an $s$-times integrated Wiener process are almost surely $s$-times differentiable. So it seems likely that achieving good performance with a Gauss-Markov-Runge-Kutta solver requires trading off the good marginal variance coverage of high-order Markov models (i.e. repeatedly integrated Wiener processes) against modelling non-smooth solution paths with lower degrees of integration. We leave this very interesting question for future work.

4 Experiments

Since Runge-Kutta methods have been extensively studied for over a century [11], it is not necessary to evaluate their estimation performance again. Instead, we focus on an open conceptual question for the further development of probabilistic Runge-Kutta methods: If we accept high convergence order as a prerequisite to choose a probabilistic model, how should probabilistic ODE solvers continue after the first $s$ steps? Purely from an inference perspective, it seems unnatural to introduce new evaluations of $x$ (as opposed to $\dot{x}$) at $t_0 + nh$ for $n = 1, 2, \ldots$. Also, with the exception of the Euler case, the posterior covariance after $s$ evaluations is of such a form that its renewed use in the next interval will not give Runge-Kutta estimates. Three options suggest themselves:

Naïve Chaining One could simply re-start the algorithm several times as if the previous step had created a novel IVP. This amounts to the classic RK setup. However, it does not produce a joint “global” posterior probability distribution (Figure 2, left column).

Smoothing An ad-hoc remedy is to run the algorithm in the “Naïve chaining” mode above, producing $N \times s$ gradient observations and $N$ function evaluations, but then compute a joint posterior distribution by using the first $s$ gradient observations and 1 function evaluation as described in Section 3, then using the remaining $s(N - 1)$ gradients and $(N - 1)$ function values as in standard GP inference. The appeal of this approach is that it produces a GP posterior whose mean goes through the RK points (Figure 2, center column). But from a probabilistic standpoint it seems contrived. In particular, it produces a very confident posterior covariance, which does not capture global error.
Continuing after s evaluations Perhaps most natural from the probabilistic viewpoint is to break with the RK framework after the first RK step, and simply continue to collect gradient observations—either at RK locations, or anywhere else. The strength of this choice is that it produces a continuously growing marginal variance (Figure 2, right). One may perceive the departure from the established RK paradigm as problematic. However, we note again that the core theoretical argument for RK methods is only strictly valid in the first step, the argument for iterative continuation is a lot weaker.

Comparison to Square-Exponential kernel Since all theoretical guarantees are given in forms of upper bounds for the RK methods, the application of different GP models might still be favorable in practice. We compared the continuation method from Fig. 2 (right column) to the ad-hoc choice of a square-exponential (SE) kernel model, which was used by Hennig and Hauberg [6] (Fig. 3). For this test case, the GMRK method surpasses the SE-kernel algorithm both in accuracy and calibration: its mean is closer to the true solution than the SE method, and its error bar covers the true solution, while the SE method is over-confident. This advantage in calibration is likely due to the more natural choice of the output scale $\sigma^2$ in the GMRK framework.

5 Conclusions

We derived an interpretation of Runge-Kutta methods in terms of the limit of Gaussian process regression with integrated Wiener covariance functions, and a structured but nontrivial extrapolation model. The result is a class of probabilistic numerical methods returning Gaussian process posterior distributions whose means can match Runge-Kutta estimates exactly.

This class of methods has practical value, particularly to machine learning, where previous work has shown that the probability distribution returned by GP ODE solvers adds important functionality over those of point estimators. But these results also raise pressing open questions about probabilistic ODE solvers. This includes the question of how the GP interpretation of RK methods can be extended beyond the 3rd order, and how ODE solvers should proceed after the first stage of evaluations.

Acknowledgments

The authors are grateful to Simo Särkkä for a helpful discussion.
References


